

Web-to-PDF

将 Markdown/HTML/URL 打印为专业 PDF 文档，适合技术文档、报告与归档。

主题风格

本技能提供 5 种精心设计的主题：

| 主题 | 风格描述 |
|----------|-----------------------|
| default | Apple 设计风格，简洁优雅 |
| github | GitHub README 风格，清晰易读 |
| academic | 学术论文风格，双栏布局 |
| sketch | 手绘笔记风格，个性化表达 |
| magazine | 杂志排版风格，首字下沉 |

代码高亮示例

JavaScript

```
// 转换 Markdown 为 PDF
import { toPdf } from './converter.js';

async function convert() {
  const result = await toPdf({
    inputPath: 'README.md',
    outputPath: 'output.pdf',
    style: 'github',
    options: {
```

```
        format: "markdown"
    }
});

console.log(`PDF 已生成: ${result.pdfPath}`);
}
```

Python

```
def fibonacci(n: int) -> list[int]:
    """生成斐波那契数列"""
    if n <= 0:
        return []
    if n == 1:
        return [0]

    sequence = [0, 1]
    while len(sequence) < n:
        sequence.append(sequence[-1] + sequence[-2])
    return sequence

# 示例调用
print(fibonacci(10)) # [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

表格示例

CLI 参数一览

| 参数 | 说明 | 默认值 |
|----------------------|------------------|-----|
| <code>--input</code> | 输入文件路径 (md/html) | 必需 |
| | | |

| 参数 | 说明 | 默认值 |
|---------------------------|----------------|-----------------------|
| <code>--output</code> | 输出 PDF 路径 | 必需 |
| <code>--style</code> | 主题风格 | <code>default</code> |
| <code>--format</code> | 输入格式 | <code>markdown</code> |
| <code>--no-html</code> | 不保留中间 HTML | <code>false</code> |
| <code>--no-mermaid</code> | 禁用 Mermaid 预渲染 | <code>false</code> |

引用块样式

提示：本技能支持 Mermaid 图表预渲染。
只需在 Markdown 中使用标准的 mermaid 代码块即可。

列表样式

无序列表

- 支持 Markdown 语法高亮
- 自动处理 Mermaid 图表
- 多种预设主题可选
- Print-ready 输出品质

有序列表

1. 安装依赖: `npm install`
 2. 运行 Playwright: `npx playwright install chromium`
 3. 执行转换: `node scripts/converter.js --input doc.md --output output.pdf`
-

注脚与强调

这是一段包含 **粗体**、*斜体*、`行内代码` 和 [链接](#) 的文本。

支持数学公式的场景可以使用 LaTeX 语法^[1]。

分隔与排版

本技能专注于 **输出品质**，每个主题都经过精心调校：

- 字体层级清晰
- 舒适的行高与间距
- 代码块语法高亮
- 表格与引用美化

高质量的 PDF 输出，从 Markdown 开始。

1. 目前需要额外配置支持 ↩